



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

09/606,786

06/28/2000

Robert Murphy

B. Carlson

6327

29906 7590 08/09/2007
INGRASSIA FISHER & LORENZ, P.C.
7150 E. CAMELBACK, STE. 325
SCOTTSDALE, AZ 85251

EXAMINER

CHOUDHURY, AZIZUL Q

ART UNIT

PAPER NUMBER

2145

MAIL DATE

DELIVERY MODE

08/09/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

09/606,786

Applicant(s)

MURPHY ET AL.

Examiner

Azizul Choudhury

Art Unit

2145

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 23 April 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1,3-25,42-90,93-101,105 and 107-116 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 1,3-25 and 42-89 is/are allowed.
- 6) ☒ Claim(s) 90, 93-101, 105 and 107-116 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 28 June 2000 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- ☐ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- ☐ Notice of Informal Patent Application
- ☐ Other: _____

DETAILED ACTION

This office action is in response to the correspondence received on April 23, 2007.

Response to Amendment

Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn.

Drawings

New corrected drawings in compliance with 37 CFR 1.121(d) are required in this application because the drawings submitted on June 2000 are hand drawn and are difficult to read when printed. Applicant is advised to employ the services of a competent patent draftsman outside the Office, as the U.S. Patent and Trademark Office no longer prepares new drawings. The corrected drawings are required in reply to the Office action to avoid abandonment of the application. The requirement for corrected drawings will not be held in abeyance.

Allowable Subject Matter

Claims 1, 3-25 and 42-89 are allowed.

The following is a statement of reasons for the indication of allowable subject matter: The claims describe an invention wherein prior to a boot, a system is able to determine the attributes of a client system. In addition, the system automatically selects

Art Unit: 2145

one of multiple instruction sets stored on a server for the client, prior to the booting of the client.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 90, 98-101, 105 and 107-115 are rejected under 35 U.S.C. 102(b) as being anticipated by Chang (US5680547A), hereafter referenced as Chang.

3. With regards to claim 90, Chang teaches a system for managing a client computer over a network, the client computer having a plurality of client computer attributes and a local operating system, the system comprising:

- A database configured to store a plurality of template records, each of said plurality of template records comprising a set of template attributes and a corresponding configuration script (Chang discloses a design which features a database within the server for holding the access control list (ACL) (column 4, line 3, Chang). If the database can hold an access control list, the access control list can contain information concerning the client as claimed. Such information is equivalent to the claimed template records. In addition, ACLs typically do contain multiple records); and

- A server application configured to receive a request from said client computer via said network, to provide a preboot attribute determination program to the client computer in response to the request, to receive said client computer attributes from the preboot attribute determination program executing on the client computer, to associate said client computer with at least one of said template records by comparing said client computer attributes to said template attributes, and to provide the configuration script corresponding to the associated at least one of said template records to said client computer for execution on said client computer prior to booting a local operating system (As stated above, Chang's design provides for communication between the server and the client machine (column 4, lines 23-24, Chang). This communication allows for the server to receive requests from the clients with information from the client (such as records) as claimed. Plus, Chang's design allows for the use of scripts to create utilities to perform administrative tasks (column 4, lines 33-35, Chang). The claimed tasks of configurations are administrative tasks. In addition, the utilities made from scripts are run on client machines and are sent through the communication (network) established between the client and server (column 4, lines 23-24, Chang). Hence the claimed step of sending programs to the client to and executed on the client exists within Chang's design).

Art Unit: 2145

4. With regards to claim 98, Chang teaches a system wherein said records of information comprise attributes of said client computers (As stated above, Chang's design consists of a server management application. For such an application to function properly, it is inherent that it not only contains the claimed management instructions, but it also should select the appropriate management instructions for each client machine. If this step did not exist, a server management application could never properly perform its task).

5. With regards to claim 99, Chang teaches a system wherein said attributes comprise DMI attributes (Chang's design has client machines with NICs to allow for interfacing with desktop server management applications (column 4, lines 13-15, Chang). Furthermore, Chang's design consists of a hardware device attached to each client machine to permit access to the client from the desktop management software at any given time (column 2, lines 38-54, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of DMI attributes as claimed).

6. With regards to claim 100, Chang teaches a system wherein said attributes comprise PCI attributes (PCI is a type of hardware and Chang's design has client machines with hardware (column 4, lines 10-12, Chang). These can be accessed by the server management application of Chang's design hence, the attributes can comprise of PCI attributes as claimed).

7. With regards to claim 101, Chang teaches a system wherein said attributes comprise SMBIOS attributes (SMBIOS is a type of BIOS. Chang's design allows for the server management application within a server to access a client's BIOS (column 2, line 48, Chang). For the server management application to access the BIOS properly, it must have some means by which to obtain the attributes of the BIOS. Since SMBIOS and BIOS are viewed as being equivalent, the server management application must be able to obtain the SMBIOS attributes).

8. With regards to claim 105, Chang teaches a system for administering a plurality of client computers over a network, the system comprising;

- Means for receiving a boot message from one of said plurality of client computers (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages);
- Means for determining attributes of said one of said plurality of client computers prior to booting a local operating system of said one of said

plurality of client computers (Since Chang's design allows for pre-boot updates, it is inherent that there exists means for obtaining attributes of clients before they boot);

- (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages. It is well known in the art that messages are simply data. The claimed received boot messages are able to contain data about the client booting. If boot messages are received then it is also inherent that the server management application should be able to detect which client machine the boot message is from (since such information will be contained within the message).
- Means for associating said attributes with an entry in a database to determine administration steps to be performed on said one of said plurality of client computers (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the

server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages. In addition, Chang discloses that the server with the server management application has a database. A database stores data in some order and hence has entries. It is thus inherent that the boot messages can be associated with entries within the database); and

- Means for providing said administrative steps to said one of said plurality of client computers for execution prior to booting the local operating system of said one of said plurality of client computers (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages).

9. With regards to claim 107, Chang teaches a system wherein said determining means comprises means for querying hardware and software attributes of one of said plurality of client computers (Chang's design permits the server with the server management application to meter and diagnose client machines (column 3, lines 31-33, Chang). Having the ability to diagnose and meter clients is equivalent to having the ability to determine a client's attributes).

10. With regards to claim 108, Chang teaches a system wherein said querying means comprises means for querying DMI parameters of one of said plurality of client computers (Chang's design has the means by which to monitor nodes, meter and diagnose software, and manage a client machine's configuration (column 3, lines 12-34, Chang). To perform such tasks, the means by which to query the client about this information must exist. Since Chang's design has the means by which to query a client, it is inherent that Chang's design also has the means by which to query a client's DMI parameters).

11. With regards to claim 109, Chang teaches a system wherein said querying means comprises means for querying PCI parameters of one of said plurality of client computers (Chang's design allows for a client machine's hard drive to be accessed remotely from the server management application (column 3, lines 21-22, Chang). To properly access the hard drive, the server management application must have means by which to query the hard drive. The hard drive like the PCI is a form of hardware. Since Chang's design allows for the querying of a hard drive, it is inherent that it must also enable the means by which to query PCI parameters).

12. With regards to claim 110, Chang teaches a system wherein said querying means comprises means for querying SMBIOS parameters of said one of said plurality of client computers (Chang's design allows the server management application within a

Art Unit: 2145

server to provide pre-boot services (column 4, lines 47-67). Chang's design also allows the server management application within a server to access the client's hard drive, meter and diagnose software and manage configurations (column 3, lines 12-34, Chang). To properly perform the tasks described, it is inherent that the means by which to query a client's BIOS. A SMBIOS is a form of BIOS and thus, Chang's design is considered to allow for the querying of a client's SMBIOS attributes).

13. With regards to claim 111, Chang teaches a method of maintaining files on a client computer having a local operating system and a network interface card, the method comprising the steps of:

- Receiving a boot request at a server computer from said client computer (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages);
- Providing a response to said boot request from said server to said client via said network interface card, wherein said response comprises a file checking program configured to be executed on said client computer prior to booting

said local operating system (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. The server management application within the server is capable of receiving boot messages. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages. In addition, Chang discloses that his design is capable of providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that a file-checking program be used to first detect if anything is missing. Plus, Chang's design allows for the use of scripts to create utilities to perform administrative tasks (column 4, lines 33-35, Chang). Means exist within Chang's design to send and execute such scripts to the client as claimed, prior to booting);

- Receiving an index of files present on said client computer from said file checking program without booting said local operating system; (As stated above, Chang's design allows for workstation crash recovery. To perform the task of crash recovery properly, it is inherent that some index of files be available to properly detect if files are missing. It would further be inherent that such an index of files be received on the client machine);
- Providing updated files from said server to said client computer based upon said index (As stated above, Chang's design allows for workstation crash recovery. To perform the task of crash recovery properly, it is inherent that

updated files be sent from the server to the client based upon an index of files); and

- Instructing said client computer to boot said local operating system after said updated files are received from said server.

14. With regards to claim 112, Chang teaches a method comprising the step of mounting a volume of said server to said client computer (Chang's design calls for both server and client machines. The design enables the client and server to communicate between each other (column 4, lines 23-24, Chang). It is inherent that both the server and client machines have drives such as hard drives. Since a client can communicate with a server (which is remote), it is possible for a client to access a server's hard drive. This accessing of a server's hard drive requires mounting and hence the claimed ability to mount a remote drive from the server to the client is viewed as being inherent).

15. With regards to claim 113, Chang teaches a method wherein said volume is mounted via a network stack located in a ROM on said client computer (Chang's design allows for a server and client to communicate amongst each other (this is a network). As stated before, Chang's design allows the client to mount a volume on the server. The claimed stack in ROM is inherent since various tasks apply stacks in the computing world and a ROM is simply a means for storage of the stack. The task of mounting a volume as claimed would qualify as one of the tasks that would inherently use a stack and ROM, hence the claim is rejected)

16. With regards to claim 114, Chang teaches a method wherein said ROM is a ROM on a network interface card of said client computer (Chang's design features a hardware component that is installed on the LAN card. A LAN card is a network interface card (NIC). The hardware component can hold programs and perform permits and helps execute some of the administrative tasks from the server management application within the server. Chang further discloses that the hardware component has ROM as claimed).

17. With regards to claim 115, Chang teaches a method wherein said ROM is a PXE-enabled ROM (PXE-enabled ROM is beneficial for its pre-boot functionality. Chang's design features ROM but also describes how the design can perform administrative tasks prior to loading of the operating system (column 2, lines 46-47, Chang). This functionality is equivalent to that of the PXE-enabled ROM. The claim is therefore rejected).

Claim Rejections - 35 USC § 103

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

invention was made to a person having ordinary skill in the art to which said subject matter pertains.

Patentability shall not be negated by the manner in which the invention was made.

19. Claims 93-94 and 116 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang in view of Havekost et al (Pat No. US005768119A), referred to hereafter as Havekost.

20. With regards to claim 93, Chang teaches through Havekost a system further comprising event objects associated with at least one of said template records, wherein said event objects are associated with said configuration scripts such that said configuration scripts are provided to said client computers upon the occurrence of an event (As stated before, Chang's design has server management applications which permit administrators to meter, diagnose and recover files (also stated earlier, software is made of files and the two are considered equivalent) within client machines (column 3, lines 30-32, Chang). To properly perform these tasks, it is inherent that the attributes of the clients being worked on should be accessible to the server management application and that the attributes must exist in the form of files (these are equivalent to the claimed records of information). Since attributes (in the form of files) are accessible to the server management application, Chang's design contains attribute files. Furthermore, Chang discloses that his design enables the usage of scripts (column 4, lines 33-35, Chang). Chang's design however fails to disclose the use of event and template objects.

Havekost discloses a design which uses event objects (column 31, lines 53-55, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost). Objects can hold information, such as the records of information claimed.

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design that associates event objects with scripts so that the scripts respond to an occurrence of an event, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

21. With regards to claim 94, Chang teaches through Havekost a system wherein said event comprises the booting of one of said client computers (Chang's design discloses how updated files can be transferred to the client by the server management application within the server (column 4, lines 47-67, Chang). As stated before, the server management application can use scripts that run whatever tasks there are programmed to. Hence, the scripts can be created to respond to an event. The booting of a client machine is an event and Chang's design describes how clients can boot (column 4 lines 47-67, Chang). Hence a script can respond to a boot process. Chang's design however fails to disclose the use of event and template objects.

Havekost discloses a design which uses event objects (column 31, lines 53-55, Havekost) and template objects (a template for an object can be a template object) (column 9, lines 21-22, Havekost). Objects can hold information, such as the records of information claimed.

It would have been obvious to one skilled in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost to create a design that associates event objects with scripts so that the scripts respond to an occurrence of an event, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

22. With regards to claim 116, Chang teaches through Havekost a method of maintaining a registry on a client computer having a local operating system and a network interface card, the method comprising the steps of:

- Receiving a boot request at a server computer from said client computer (Chang discloses a design that allows for server and client machines to communicate with each other (column 4, lines 19-24, Chang). Chang further discloses in his design that the server management application within the server can offer pre-boot updates and allow the client to access the server while the client is booting (column 2, lines 38-67, Chang). With such abilities

available in the design, it is inherent that the server management application within the server is able to detect when the client is booting and thus is able to receive boot messages. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to receive a boot request from a client maintaining a registry, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- Providing a registry checking program to the client computer via the network interface card in response to said boot request, wherein said registry checking program is configured to check said registry on said client computer prior to booting said local operating system and to provide a registry response to said server computer via said network interface card. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to respond to a boot request from a client maintaining a registry, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- Receiving said registry response at said server from said registry checking program (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. The server management application within the server is capable of receiving boot messages. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages. In addition, Chang discloses that his design is capable of providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that a registry-checking program be used

to first detect if anything is missing. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to receive a registry program from a registry checking program, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang));

- Processing said registry response at said server to verify said registry on said client computer (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. The server management application within the server is capable of receiving boot messages. Since boot messages can be received, it is inherent that the administrative steps can be performed in response to boot messages. In addition, Chang discloses that his design is capable of

providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that the server be able to verify the registry. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server verify the registry, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)); and

- Providing an updated registry from said server to said client computer in response to said processing step (As stated before, Chang's design allows for boot messages and client computers. The server in Chang's design contains a server management application that is used for performing administrative tasks on client machines. The server management application within the server is capable of receiving boot messages. Since boot messages can be

received, it is inherent that the administrative steps can be performed in response to boot messages. In addition, Chang discloses that his design is capable of providing crash recovery (column 3, line 30, Chang). To properly perform the task of crash recovery, it is inherent that the server provides an updated registry to the client. Chang's design however fails to describe the role of the registry file (registry).

Havekost's design discloses the use of a registry database (column 31, line 47, Havekost). A database stores files and hence the existence of registry files is taught. Registry files exist within computers, and both clients and servers are computers.

Thus it would have been obvious to one skilled in the art at the time of the invention to have combined the teachings of Chang with those of Havekost to permit a server to provide a updated registry to the client, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

- Instructing said client computer to boot said local operating system after said updated registry received from said server.

Art Unit: 2145

23. Claims 95-97 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chang in view of Havekost et al (Pat No. US005768119A), referred to hereafter as Havekost and in further view of Sonderegger et al (Pat No. US005692129A), referenced hereafter as Sonderegger. Chang discloses a design with server and client machines that can communicate between one another (column 4, lines 23-24, Chang). In addition, Chang teaches how the client files can be checked by the server management application within the server. (In column 7, line 47-53, Chang discloses that the client can be checked to ensure the software is up to date. Software is made of files or could consist of only one file; hence software and files are viewed as being equivalent. Additionally, Chang states in column 3, lines 30-32 how his design allows for workstation recovery, metering and diagnostics; hence the design allows for checks, recovery, metering and diagnostics of files.) Such capabilities make it inherent that an index of some form must exist to enable client files to be checked and recovered. Chang however fails to specify the role of objects and directory service applications within his design.

24. In the same field of endeavor, Havekost discloses a design that uses registries (column 31, line 46-48, Havekost), event objects (object that handle events are viewed as being equivalent to event objects, column 31, lines 53-55, Havekost), template objects (a template that serves as a template for an object is viewed as being equivalent to a template object, column 9, lines 21-22, Havekost) and workstation objects (an attribute object for a client is viewed as being equivalent to a workstation object, column

Art Unit: 2145

25, lines 4-10, Havekost). Havekost's design however fails to describe the role of directory services applications.

25. Sonderegger discloses a design which features a directory services application (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database.

26. Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang).

27. With regards to claim 95, Chang teaches through Havekost and Sonderegger a system wherein said database is a directory services application (Chang design teaches that databases can be used (column 4, line 3, Chang). Chang however fails to describe the role of objects and directory services applications.

In the same field of endeavor, Havekost discloses a design that features event objects (column 31, lines 53-55, Havekost), template objects (column 9, lines 21-22, Havekost) and workstation objects (column 25, lines 4-10, Havekost). Havekost however does not disclose details about a directory services application.

Sonderegger discloses a design which features a directory services application (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database.

Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang).

28. With regards to claim 96, Chang teaches through Havekost and Sonderegger a system wherein said directory services application is a Netware Directory Services™ directory (Chang design teaches that databases can be used (column 4, line 3, Chang). Chang however fails to describe the role of objects and directory services applications.

In the same field of endeavor, Havekost discloses a design that features event objects (column 31, lines 53-55, Havekost), template objects (column 9, lines 21-22, Havekost) and workstation objects (column 25, lines 4-10, Havekost). Havekost however does not disclose details about a directory services application.

Sonderegger discloses a design which features Netware Directory Services (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database.

Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang).

29. With regards to claim 97, Chang teaches through Havekost and Sonderegger a system wherein said directory services application is a Microsoft Active Directory™ directory (Chang design teaches that databases can be used (column 4, line 3, Chang). Chang however fails to describe the role of objects and directory services applications.

In the same field of endeavor, Havekost discloses a design that features event objects (column 31, lines 53-55, Havekost), template objects (column 9, lines 21-22, Havekost) and workstation objects (column 25, lines 4-10, Havekost). Havekost however does not disclose details about a directory services application.

Sonderegger discloses a design which features a directory services application (NDS is a type of directory services application, column 5, lines 14-15, Sonderegger). Such a directory services application can serve as a database. Microsoft Active Directory like NDS is a directory services application. Hence, the use of one can be substituted by the use of the other.

Accordingly it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Chang with those of Havekost and Sonderegger, for the purpose of creating a method and apparatus for maintaining a computer system to enable the workstation to communicate with a server on the network and make the necessary resources of the workstation available to a server management application running on the server via the network (column 2, lines 49-52, Chang)).

Response to Remarks

The correspondence received on April 23, 2007 has been considered but is not deemed fully persuasive. In lieu of the arguments presented, claims 1, 3-25 and 42-89 have been deemed allowable however; the remainder of the claims remains rejected. Applicant's arguments with respect to claims have been considered but are moot in view of the new office action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Azizul Choudhury whose telephone number is (571) 272-3909. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jason Cardone can be reached on (571) 272-3933. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2145

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

AC



JASON CARDONE
SUPERVISORY PATENT EXAMINER